Mathematica: A System for Doing Mathematics by Computer, Second Edition. *By Stephen Wolfram.* Addison-Wesley, Redwood City, California, 1991. xxii + 961 pp. ISBN 0-201-51502-4.

The introduction of Mathematica has done more to stir passions in the field of symbolic mathematics than has the entry of any other comparable system since the 1960's. This has proved to be a good thing because the controversy has brought integrated symbolic, numeric and graphical systems to the forefront of awareness, both in the realms of science and education. People the world over have become much more cognizant of the inherent possibilities that such a system can provide in aiding their researches and their understanding of mathematical phenomena.

Moreover, the introduction of Mathematica has encouraged an accelerated development of other integrated mathematical software, and this has in turn spurred a more general awareness of the variety of existing symbolic packages. These include Derive (a successor of muMATH), MACSYMA, MAPLE, REDUCE and Scratchpad II (now being marketed as Axiom), to name a few of the more prominent systems. There are a host of others that tend to be less sophisticated and/or more specialized in their purpose. Any potential user would be wise to check out several of these systems and see which one (or ones) are the most suitable. A number of reviews have appeared in the **Computers and Mathematics** column of the *Notices of the American Mathematical Society*. For those with USENET access, interesting tidbits appear from time to time in the newsgroup sci.math.symbolic. There are also electronic mail discussion lists, including one just on Mathematica (mathgroup@yoda.physics.unc.edu).

The review copy of Mathematica that was provided by Wolfram Research was Version 2.0.2 for a Macintosh with a 68020/30 chip and a numeric coprocessor. 4 megabytes of memory are required, but to do anything at all fancy, this number should probably be doubled. The collection of items that comes with this version includes 15 disks (6 for the program, 6 containing sample notebooks and 3 of miscellaneous utilities), a spiral bound **User's Guide For the Macintosh**, a **Guide to Standard Mathematica Packages** (some of the pages of which were interleaved incorrectly in the copy provided), and additional booklets, licenses, T-shirt ads, etc. In addition to the Macintosh, I also have had experience running Mathematica on various UNIX workstation platforms.

Mathematica appears to be like the little girl with the little curl right on the middle of her forehead: when it is good, it can be very, very good, but when it is bad, it can be horrid. For example, one of Mathematica's strongest features is its graphics. It is easy to produce density, contour and 3-D plots of a surface and to interconvert between any of these forms. On the Macintosh, sequences of plots can be animated by repeated cycling, making for an extremely nice visualization tool. All graphics are done in PostScript which allows plots to be incorporated easily into other programs' formats (like TEX output using an appropriate device driver). A new feature in Version 2 allows sounds to be also treated as graphics objects, and so one can "play" a one parameter function.

On the other hand, Mathematica seems not to perform a singularity analysis for definite integrals except at the endpoints, so that spurious results like $\int_{-1}^{1} 1/x\,dx$ yielding $-i\pi$ can occur unexpectedly (this was such a widespread complaint in the previous version of Mathematica that Wolfram added a note about this problem in the current book). Another weak area is linear algebra. Asking for the eigenvalues of a diagonal matrix with atomic symbolic entries in the diagonal is exceedingly slow. For instance, the 9×9 case required 78 seconds on a Sun IPC.

One of the most innovative features of Mathematica is its user interface, in particular, the notebooks found in the Macintosh and NeXT versions. A notebook allows one to mix text, graphics and mathematics in an interactive textbook format. Various levels of sectioning are available, in which details of a topic are hidden from view unless specifically requested by a mouse click. For purposes of demonstration, notebooks can be very illuminating. (It is ironic that in the sample notebooks provided, the inputs used to create some of the more interesting graphical objects are omitted.) One difficulty is the lack of ability to include typeset mathematics.

Another nice feature of the user interface is the online help which allows one to use UNIX style wildcards to help track down the name of a function. However, this only works initially for functions in Wolfram's book, even though the system is supplied with "standard" packages. Moreover, no examples of function usage are provided, something which both MACSYMA and MAPLE have as standard items. The Mathematica book does do a good job in this regard, but one does not always have it handy.

Correcting interactive input is difficult in a non-windowing UNIX environment (once edited input cannot be recalled), although it is easy enough to work with files. On the Macintosh, on the other hand, standard mouse oriented editing is available on the command history. Mathematica tries to intuit when an input line ends. This is all right for short commands, but one must always be careful with multiline input when breaking lines, or resort to surrounding everything in parentheses (a strategy suggested by Wolfram) to ensure that the input is not terminated prematurely. (One sometimes needs to use this trick to make output generated by Mathematica's `Write` command read back in correctly.)

The Mathematica language is interesting. It is established on a combination of functional and rule-based ideas. The elements are strongly grounded on pattern matching in which side conditions are allowed. Patterns can be rather general and pattern variables may be strongly typed if desired. The syntax tends to be compact and is fairly uniform. Many C and LISP constructs are available. Like C, Mathematica syntax is very sensitive to single character changes, which can sometimes be difficult to spot.

As an example of the language, $n!$ can be defined recursively by

```
f[0] = 1
f[n_Integer /; n > 0] := n f[n-1]
```

Here, the `=` implies immediate assignment to the indexed variable `f`, while the `:=` indicates a computation by the function `f`. Mathematica tries to apply more specific rules first, although for more complicated definitions, it cannot often decide; however, the user may with great care explicitly manipulate the order of rule application. One must also be careful not to place too many rules on an operator, especially a common one like `Plus` or `Times`, as this can substantially slow down calculations due to the need to perform several pattern matches every time the operation takes place. The ability to associate a rule with a secondary level operator, e.g., `g` in

```
g/: h[x_] + g[y_] := hg[x, y]
```

can help to alleviate overloading, although one must watch for situations such as replacing the sum by a division since `h[x]/g[y]` is represented internally by `Times[h[x], Power[g[y], -1]]`, and so `g` would be a tertiary level operator in this case.

The Mathematica language has a great deal of flexibility and good programming methods are (usually) encouraged. It is certainly quite possible to write large packages in a sophisticated manner with internal functions localized by context hiding, a feature not available in most other computer algebra systems. The behavior of the language, however, can be quirky and so one must be careful. For example, bugs where local variables act in a non-local manner have been noted recently by posters in sci.math.symbolic.

Mathematica implements a great number of functions and tries to cover a wide breadth of pure mathematics along with numerical and data analysis in its quest to be a comprehensive tool. This provides a nice framework for experimentation, but the depth of coverage of topics tends to be rather spotty. Derivatives are implemented more cleanly than in many other systems, which often have had trouble in the past distinguishing partial derivatives from ordinary derivatives. A large number of orthogonal polynomials and special functions of physics are conveniently provided. Also, there are a variety of procedures for manipulating statistical data available in the packages.

Mathematica fails, however, to provide standard methods for automatic algebraic simplification such as the canonical rational expression representation for polynomials and rational functions that MACSYMA possesses. One can program around this to some extent, but at a loss in efficiency.[1] Branches of complex functions are chosen inconsistently. In the last 10 tests posed by Stoutemyer [2], `PowerExpand` would typically simplify too much while `Simplify` would not go far enough when applied to expressions containing fractional powers, logarithms, trigonometric functions and inverse trigonometric functions. (To be fair, computer algebra systems in general have problems in this area.) The generation of FORTRAN code is not that good, being much inferior to the results produced by GENTRAN [3]. With respect to efficiency, facilities for compilation of user code (except strictly numerical segments) are missing, and memory management is often not very effective on large problems (notebooks or even the entire system would suddenly crash on the Macintosh).

Mathematica can produce numerical results for many operations (e.g., definite integration, solutions of ODEs, roots of equations, etc.). The algorithms employed seem to work well, at least when using machine precision numbers. For example, Mathematica whipped out a reasonable solution to a rather delicate system of nonlinear ODEs that had given other purely numerical programs fits (this problem was suggested by Nicholas Kazarinoff). However, the situation is less clear with arbitrary-precision numbers. Mathematica defines the `Accuracy[x]` as the number of digits to the right of the decimal point in `x`, while the `Precision[x]` is the number of "significant" digits in `x`. Results are computed keeping track of these two quantities. These definitions are contrary to what most other people use (for example, see [4]) and can lead to surprising behaviors. The implementation is suspect, also, since by replacing a number by the arithmetic mean of n copies of it ($n \geq 4$) some large number of times (say 100), one can increase the `Precision` and `Accuracy` arbitrarily!

The documentation for Mathematica is particularly helpful. The Mathematica book is well laid out, typeset in TeX to be very readable and thoroughly indexed. Function syntax is boxed in gray for emphasis, below which examples of usage with marginal annotations are typically given. The book begins with some impressive examples of Mathematica in action (including a gallery of pretty color plots—again, unfortunately, without the input that

[1] Rolf Mertig, in his high energy physics package FeynCalc [1], has written some functions (e.g., `Factor2`) that emulate portions of MACSYMA's behavior in this regard.

generated most of them). This is followed by an introduction to the basic features of Mathematica, after which the principles underlying the language are explained in great detail. The next part discusses advanced mathematical operations such as algebraic manipulation, calculus, power series and so forth. The final section of the book, provided as an appendix, is alphabetically arranged by function and option name for easy reference. Each entry includes a short description, and also furnishes a pointer to other relevant entries and a page reference to the appropriate discussion located in the main portion of the text. All computer manuals should be laid out so well. The Mathematica book does have a few irritations, though. The functions in the standard packages are sometimes mentioned but are not really integrated into the text (for instance, they are not listed in the reference manual). The author often implies that if Mathematica cannot solve a problem then probably no other system can either, which is clearly misleading for someone who has had little experience with symbolic math programs. Finally, virtually no details of algorithms used are given nor are any references provided. Such a black box approach should make any scientifically oriented person cringe.

In summary, Mathematica 2.0 has many features to commend it: graphics, sound, notebooks and language, among others. User applications have been developed in a number of diverse areas, such as high energy physics [1], life sciences, education and finance. However, Mathematica is no panacea despite the hype and the wary user should take the time to also check out the competition.

I would like to thank Rolf Mertig for several helpful comments as I was writing up this review.

REFERENCES

[1] R. Mertig, M. Böhm and A. Denner, "FeynCalc—Computer-algebraic calculation of Feynman amplitudes", *Computer Physics Communications* **64** (1991) 345.

[2] David R. Stoutemyer, "Crimes and Misdemeanors in the Computer Algebra Trade", *Notices of the American Mathematical Society*, Volume 38, Number 7, September 1991, 778–785.

[3] Barbara L. Gates, "A Numerical Code Generation Facility for REDUCE", *Proceedings of the 1986 Symposium on Symbolic and Algebraic Computation*, Association for Computing Machinery, 1986, 94–99.

[4] Stanley Wolf and Richard F. M. Smith, *Student Reference Manual for Electronic Instrumentation Laboratories*, Prentice Hall.

MICHAEL WESTER
University of New Mexico